31.1.2024 Rehuven Tesler

# License Frontline API package (LicFls)

## Background & motivation:

As part of Frontline IP protection, our standalone services should be able to run only when a dongle exists in the environment (network). Additional protection is to check / lock licenses so the service will could not transfer to different customers but only these that have relevant licenses.

LicFls implemented to answer these requirements and once the service compiled and linked to this package it will be able to execute only with dongle and the service will be able to check/lock licenses.

By default, LicFls will check specific license based on the type of the service want to use.

Technical:

Current version of LicFls supporting only LDK driverless method (in the future we plan to extent it to use other methods).

Internally this package checking existence of specific license to run.

Developer can use the API’s using license ID or license NAME (7 characters)   
When using license ID, no need to update the package and it enough to have the license generated in the dongle.  
When using license NAME it should be hard-coded map inside the package so if a developer wants to use license NAME and he have a new license we are generating for customer. It should be added to the code and package need to be re-compiled.

API’s:

1. **lic\_ldk\_fls\_init\_server(const char\* serverName, lic\_ldk\_fls\_type t)**  
   Explanation: initialize the package with the relevant server and license type.

Parameters:

* serverName – can be machine name of IP of the license server
* t - license type we want to use:  
   0 – CAM : checking inprofl existence   
   1 – PS : checking inpro50 existence (temporary)

1. **lic\_ldk\_fls\_lock\_id(int licId)  
   lic\_ldk\_fls\_lock\_name(const char\* licName)**  
   Explanation: lock license

Return value: error code (0 for OK)

Parameters:

* licID – license ID or
* licName - license name (required hard coded map)

1. **lic\_ldk\_fls\_unlock\_id(int licId);**

**lic\_ldk\_fls\_unlock\_name(const char\* licName);**  
  
Explanation: unlock (release) license – will work only if license locked by same instance, otherwise will return an error.

Return value: error code (0 for OK)

Parameters:

* licID – license ID or
* licName - license name (required hard coded map)

1. **bool lic\_ldk\_fls\_exists\_id(int licId);**

**bool lic\_ldk\_fls\_exists\_name(const char\* licName);**

Explanation: check license exitance – will not occupy a license but will check if a valid license exists in the dongle

Return value: true – if exists, false – if not exists/expired

Parameters:

* licID – license ID or
* licName - license name (required hard coded map)